CISTER
Research Center in
Real-Time & Embedded
Computing Systems

Conference Paper

Response Time Analysis of Slotted WiDOM in Noisy Wireless Channels

Maryam Vahabi*

Stefano Tennina

Eduardo Tovar*

Björn Andersson

*CISTER Research Center

CISTER-TR-150703

2015/09/08

Response Time Analysis of Slotted WiDOM in Noisy Wireless Channels

Maryam Vahabi*, Stefano Tennina, Eduardo Tovar*, Björn Andersson

*CISTER Research Center

Polytechnic Institute of Porto (ISEP-IPP)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail: mamvi@isep.ipp.pt, sota@isep.ipp.pt, emt@isep.ipp.pt, baa@isep.ipp.pt

http://www.cister.isep.ipp.pt

Abstract

Timely delivery of critical traffic is a major challenge in industrial applications. The Wireless Dominance (WiDOM) medium access control protocol offers a very large number of priority levels to suit time sensitive application requirements. In particular, assuming that its overhead is properly modeled, WiDOM enables an accurate evaluation of the network response time in the wireless domain, through the power of the schedulability analysis, based on non-preemptive and staticpriority scheduling. Recent research proposed a new version of WiDOM (dubbed Slotted WiDOM), which offers a lower overhead as compared to the original version. In this paper, we propose a new schedulability analysis for Slotted WiDOM and extend it to handle message streams with release jitter. In order to provide a more accurate timing analysis, the effectof transmission faults must be taken into account. Therefore, in our novel analysis we consider the case where messages are transmitted in a realistic wireless channel, affected by noise andinterference. Evaluation is performed on a real test-bed and the results from experiments provide a firm validation of our findings.

Response Time Analysis of Slotted WiDOM in Noisy Wireless Channels

Maryam Vahabi*, Stefano Tennina*†, Eduardo Tovar* and Björn Andersson‡
*CISTER/INESC-TEC, ISEP, Polytechnic Institute of Porto, Porto, Portugal
†WEST Aquila S.r.l., L'Aquila, Italy
‡Software Engineering Institute, Carnegie Mellon University, Pittsburgh, USA
Email: {mamvi,emt}@isep.ipp.pt, tennina@westaquila.com, baandersson@sei.cmu.edu

***Abstract*—Timely delivery of critical traffic is a major challenge in industrial applications. The Wireless Dominance (WiDOM) medium access control protocol offers a very large number of priority levels to suit time sensitive application requirements. In particular, assuming that its overhead is properly modeled, WiDOM enables an accurate evaluation of the network response time in the wireless domain, through the power of the schedulability analysis, based on non-preemptive and static-priority scheduling. Recent research proposed a new version of WiDOM (dubbed Slotted WiDOM), which offers a lower overhead as compared to the original version. In this paper, we propose a new schedulability analysis for Slotted WiDOM and extend it to handle message streams with release jitter. In order to provide a more accurate timing analysis, the effect of transmission faults must be taken into account. Therefore, in our novel analysis we consider the case where messages are transmitted in a realistic wireless channel, affected by noise and interference. Evaluation is performed on a real test-bed and the results from experiments provide a firm validation of our findings.**

***Index Terms*—Real-Time Systems, Fixed-Priority Scheduling, Medium Access Control Protocol, Wireless Communications.**

I. INTRODUCTION

Wireless communication is spreading in embedded computer systems and is enabling many future applications such as factory automation and collaborative robotics as well as inter-vehicle communication and smart buildings. These applications tend to have real-time requirements: outdated data are usually irrelevant and may degrade the quality of control. The research community has already created solutions to verify the timeliness properties of various types of computing systems. The most well-known solution is the Generalized Rate-Monotonic Analysis (RMA) [1], [2] which allows designers to prove in advance that all deadlines are met at run-time. This analysis has matured into a fully-fledged theory for single-processor computer systems and for a wired communication channel. However, it is not well-developed for wireless networks — not even for a wireless network in a single broadcast domain (a network where each node can hear and interfere with every transmission) with predictable noise patterns.

Indeed, the design of a real-time and reliable wireless Medium Access Control (MAC) protocol that is able to efficiently handle event-driven (sporadic) messages with bounded response time needs the following requirements to be met:

978-1-4673-7929-8/15/$31.00 © 2015 IEEE

(**R1**) A prioritized MAC protocol should exist for a wireless channel, so that among the transmitting nodes, the channel access is granted to the node with the highest priority.

(**R2**) The overhead related to the arbitration of the prioritized MAC protocol should scale with the number of priority levels.

(**R3**) A schedulability analysis should exist for the prioritized MAC protocol.

(**R4**) The schedulablity analysis should take into consideration the need for the retransmissions for the corrupted messages (especially in noisy channel).

(**R5**) In order to obtain an efficient system, the overhead related to the arbitration of the prioritized MAC protocol should be low.

Unfortunately, the current state of the art cannot fulfill all these requirements. The Controller Area Network (CAN) [3], designed and well investigated for wired channels, offers many priority levels (fulfilling R2). For the wireless version of CAN (WiDOM [4], [5]) there exists a schedulability analysis [6] (fulfilling R1, R2 and R3), but the existing analysis is based on the assumption that no errors occur during the message transmission (missing R4). Another issue is that this protocol imposes a large overhead (missing R5). On this account, a new version of WiDOM, called *Slotted WiDOM* [7], has been proposed to offer low overhead (fulfilling R1, R2 and R5), but no schedulability analysis is available for it. Hence, the development of a schedulability analysis for Slotted WiDOM, capable to predict the timing of those message streams suffering from jitter or experiencing noise on the channel, would help in fulfilling all the above requirements.

This paper aims to fill such a gap by providing the schedulability analysis for Slotted WiDOM. In addition, we introduce a technique for error recovery, such that the reliability of WiDOM protocol is improved under poor channel conditions. The choice of using the deterministic noise patterns allows deriving the response time analysis of message streams in a closed form. Obviously, in some cases, this is a simplification, but it also reproduces some realistic noise conditions as described in [8]. In this paper, we validated our findings with a real experimental testbed. Certainly, some other realistic scenarios would require the consideration of non-deterministic noise models. Although not in the scope of this paper, the consideration of those non-deterministic noise models would require adapting our response time analytical framework using stochastic approaches, such as those described in [9].

Accordingly, the remainder of this paper is organized as follows. Section II overviews relevant related work in the area of real-time communications for wireless networks. Section III describes the proposed error recovery mechanism for the Slotted WiDOM and the wireless channel model. Section IV details the response time analysis for Slotted WiDOM, which accounts for the errors in the transmissions. Section V validates our developed analysis through a set of extensive experiments. Finally, Section VI concludes our paper.

II. Related Work

Wireless technologies have recently gained momentum in the field of industrial control [10]–[14]. The rationale is that, in contrast to the wired control systems, their wireless counterparts offer cost effectiveness and ease of installation. Moreover, thanks to mobility support, they enable more sophisticated automation applications [10]. However, low power wireless technologies also pose new challenges.

Guaranteeing timely delivery of critical traffic in industrial applications is a major challenge. Many research works have focused on minimizing the energy consumption of these hardly resource-constrained networks. However, the real-time support to high priority traffic within such applications has started to be investigated only recently. Some attempts to support time guarantees in conventional protocols by assigning priorities to the network traffic are reported in [12], [15], [16].

The definition of the WLAN standard, IEEE 802.11 [17], stimulated the development of many prioritized Carrier Sense Multiple Access (CSMA) schemes. Some of these variants, [15], [18], [19], propose to fine tune the parameters defined in the standard in order to guarantee that deadlines are met for each traffic flow. Some others, e.g., [11], suggest to combine Time Division Multiple Access (TDMA) approaches with a traffic scheduler. Nevertheless, unless some additional scheduling features, such as those described in [20], are used, these techniques suffer from two drawbacks: (*i*) they do not fully achieve a priority-based scheduling, since it may happen that a high priority message has to wait in the transmission queue for a lower priority message to be sent; and (*ii*) collisions may still occur, as a result of using CSMA in the IEEE 802.11. Both problems certainly lead to deadline misses in real-time applications.

Due to collisions, nodes may suffer from long access delays before being able to transmit. A conventional method is to send jamming signals (called black bursts, BB) in order to win access for the transmission of higher priority messages [16], [21]. With the BB-based approach, the collision interval between two or more pulses of energy is measured in order to resolve the contention. The longest jamming transmission wins the channel access. After transmitting the BB signal, a node waits for an observation time to check if any node is transmitting a longer BB. If the channel remains idle during this time, the node sends its data frame. A recent work [12] proposed the use of BB signals to distinguish the highest priority and critical control packets. The drawbacks of BB signals lie in (*i*) the inability to cope with the hidden terminal problem, and (*ii*) the unreliability in the presence of interference.

The IEEE 802.15.4 [22] standard provides the Guaranteed Time Slot (GTS) reservation mechanism within a superframe for soft real-time traffic. The 15.4e Working Group [23] extended the standard to support the emerging needs of industrial applications. The deterministic and synchronous multichannel extension (DSME) mechanism is a solution proposed to provide prioritized channel access, by reserving DSME GTSs to the high priority traffic. However, according to this method, high priority messages are not transmitted upon their generation, but devices wait for dedicated slots to send them. Consequently, the priority inversion problem is not solved. Moreover, the GTS slots are guaranteed only upon a previous reservation request, which follows the CSMA paradigm. Thus, unless a specific mechanism such as the one described in [24] is adopted, there is no guarantee that the highest priority message will always be sent on time.

WirelessHART [20], ISA100.11a [25] and WIA-PA [26] are renowned industrial standards which leverage on the IEEE 802.15.4 physical layer and utilize TDMA mechanisms to provide GTS for time-critical transmissions. However, these standards support a limited number of priority levels (four).

The wireless dominance-based protocol (WiDOM) [5], [7] is the adaptation of the CAN bus [3] into the wireless domain and supports a very large number of priority levels. Moreover, assuming to know the main sources of interference, WiDOM allows to check in advance whether all the deadlines for the time-critical messages can be met. In this paper, we leverage on the latest version of this protocol as a baseline to extend the schedulability analysis to the case of unreliable channel conditions, i.e., nodes experiencing noise and interference, which is often the case of harsh industrial environments.

III. Ack-enabled Slotted WiDOM

Besides the physical aspects related with signal propagation in wireless media, communication errors may result from the interference generated when multiple transmitters share the same frequency spectra, like in the case of Wi-Fi with Bluetooth. To minimize the impact of such external interferers on the reliability of the WiDOM protocol, we propose to enhance Slotted WiDOM with an acknowledgment (Ack)-based error recovery scheme, i.e., a confirmation sent back to the sender after each received data packet. According to this method, if the sender does not receive an Ack packet for a predefined period of time, it will retransmit the data, provided that there is no higher priority message enqueued in the meanwhile.

A. Protocol Insights

As stated earlier, Slotted WiDOM is a younger version of WiDOM aimed to reduce the large overheads of the original version. To this end, in [7] the authors have presented a brand new add-on platform, called WiFLEX, and a novel synchronization mechanism, based on out-of-band signaling [27].

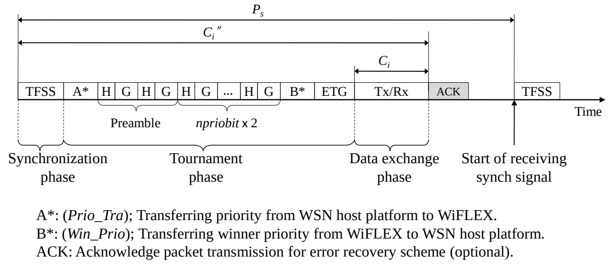


Fig. 1. Timing of Ack-enabled Slotted WiDOM (intervals are not on scale).

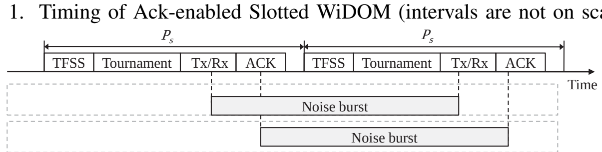


Fig. 2. Error caused by a noise burst with duration of one superframe slot.

A special node (a master node) broadcasts synchronization (*synch*) pulses on a dedicated radio channel with a periodicity P_s. These pulses lead to the definition of a superframe structure, which is shown in Fig. 1. Slotted WiDOM includes three phases: synchronization, tournament and data exchange. All of them occurring within a superframe. Synchronization provides a common reference point in time to all nodes, so that they can start the contention phase simultaneously. After synchronization, nodes contend for the channel in the tournament phase; that is, a conflict resolution phase similar to the dominance/binary countdown arbitration [3], [5], [28]. The shaded ACK box in Fig. 1 includes the switching time (from send to receive mode) and also the time a node needs to wait to receive back the Ack message from the recipient. The temporal variables in Fig. 1 are defined as follows: TFSS is the time span needed to recognize the synch signal; Prio_Tra and Win_Prio are time spans needed for the WiFLEX and host platform communications [27]; *G* is a guard time to facilitate the distinction between consecutive *H*-length priority bits; ETG is a gap at the end of the tournament to let nodes set their radio according to the result of the tournament; SWX is the time needed by the radio to switch from receive to transmit mode, or vice versa; and ACK is the duration of an an Ack packet's transmission. Hence, the value of P_s needs to be such that the message with the longest transmission time (C_i) can be fully sent and the Ack packet received, before the start of the next synch signal. This constraint is formulated as follows:

$$\begin{aligned} P_s \geq \text{TFSS} + \text{Prio\_Tra} + 2 \times (H+G) \times (\text{npriobits} + 1) \\ + \text{ETG} + \text{Win\_Prio} + \max(C_i) + \text{SWX} + \text{ACK} \quad (1) \end{aligned}$$

B. Noise Overhead Estimation

A random noise burst can cause a transmission error on either a data or an Ack packet. If such a transmission error occurs, packet retransmission is required, leading to an increase in the message transmission time. In this section, we estimate the time overhead imposed by such transmission errors.

Let δ denote the duration of a noise burst. The number of slots affected by a δ-duration noise burst is at most $1 + \lceil \frac{\delta}{P_s} \rceil$. Then the new message transmission time is delayed by $D(\delta)$ as follows:

$$D(\delta) = P_s \times \left(1 + \left\lceil \frac{\delta}{P_s} \right\rceil \right). \quad (2)$$

Fig. 2 shows an example of a noise burst with duration of one timeslot ($\delta = P_s$) that can add a delay of up to $2P_s$ on the message transmission time.

Given $F(\Delta t)$ an error function modeling the interference in a wireless channel during an interval Δt, the maximum incremental delay due to the error recovery scheme is:

$$E(\Delta t) = D(\delta) \times F(\Delta t). \quad (3)$$

As previously mentioned, we consider two different types of noise sources as proposed in [8]: (*i*) a periodic noise burst with the period of T_p and the burst duration of δ_p; and (*ii*) a sporadic noise burst with a minimum inter-arrival time of T_s and a burst duration of δ_s. The sporadic noise models the interference caused by packet-based radios (e.g., Wi-Fi, Bluetooth) while the periodic noise models the interference induced by other electromagnetic noise sources. So, being K the number of periodic noise sources and J the sporadic ones, the overall delay within the interval Δt due to the error recovery process is given by [8]:

$$\begin{aligned} E(\Delta t) = & \sum_{n=1}^{K} \left(\left\lceil \frac{\Delta t}{(T_p)_n} \right\rceil \times D((\delta_p)_n) \right) \\ & + \sum_{n=1}^{J} \left(\left\lceil \frac{\Delta t}{(T_s)_n} \right\rceil \times D((\delta_s)_n) \right). \quad (4) \end{aligned}$$

IV. Response Time Computation

The schedulability analysis presented in this section builds on a previously published analysis for CAN [29] to provide feasibility tests, based on the computation of the Worst-Case Response Time (WCRT) of messages. However, our analysis slightly differs from the one in [29], as it deals with: (*i*) the slotted nature of the protocol; (*ii*) the error detection, by including Ack-based mechanism and packet retransmissions (rather than redundancy of the symbols [29]); and (*iii*) the jitter, which generally occurs when messages are enqueued.

The WCRT of a message stream (R) is the longest response time for any message instance *q* entering the queue for a period of time called *level-i busy period (BP)*. A level-*i* BP is a time interval $[t_0, t_1)$, such that both t_0 and t_1 are the beginning of a non-faulty superframe (i.e., superframes where all the three phases perform successfully). Fig. 3 shows an example of a level-*i* BP. In particular, for each superframe in $[t_0, t_1)$ it necessarily holds that either (*i*) all transmitted packets have higher priority than m_i, or (*ii*) at most one packet with lower priority than m_i is sent in the first superframe (at t_0).

Consequently, according to [29], the WCRT of an instance q of a message stream m_i can be divided into four components:

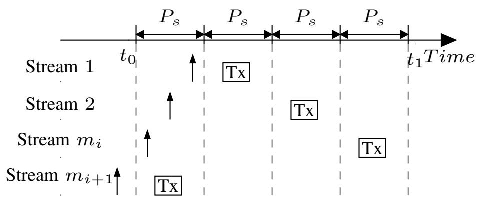


Fig. 3. Level-*i* busy period; lower index shows higher priority. The upward arrow indicates the generation time of the message.

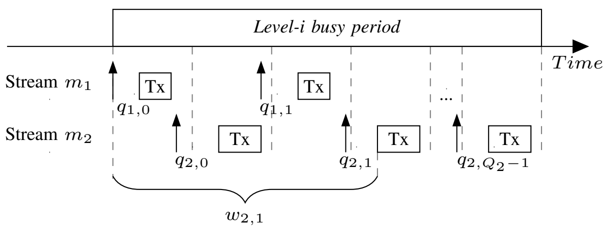


Fig. 4. An example of $\mathbf{w}_{i,q}$.

1) $w_{i,q}$: the interval from the start of the BP until when the instance q starts a successful transmission (see Fig. 4);
2) C''_i: the time span needed to finish a transmission, which includes the tournament duration and the time needed to detect a synch signal (see Fig. 1), and is given by:

$$C''_i = \mathrm{TFSS} + 2 \times (H+G) \times (\mathrm{npriobits}+1) + \mathrm{Prio\_Tra} + \mathrm{ETG} + \mathrm{Win\_Prio} + C_i \quad (5)$$

3) J_i: the release jitter (or queuing jitter [7]), which is defined as the largest difference between the initiating time of an event and the instant when the message, triggered by the event, has been enqueued;
4) E: the error overheads, including the time for receiving Ack packets, the retransmission time and the duration of the noise burst provoked by a source of interference.

Therefore, to compute the response time of a message stream m_i in Slotted WiDOM, we consider a BP of interval $[t_0, t_1)$. According to the definition, there should be a packet transmission in every non-faulty timeslot during the interval $[t_0, t_1)$. According to the release time of messages in the higher or lower priority sets, eight distinct cases must be considered in the analysis, as summarized in Table I. In the remainder of this section, we derive analytical formulations of the response time, R_i, for each of those cases. The largest response time among all $R_i^{C\ell}$, with $\ell \in \{1, 2, ..., 8\}$ will be the WCRT of message stream m_i.

*a) **Case C1**:* There is no message generated in the $[t_0 - P_s, t_0)$ interval, which is the last timeslot immediately before the level-*i* BP begins. This is the simplest case, where there is neither blocking from the *lp*(i) set nor further interference from the *hp*(i) set of messages. Thus, the WCRT for m_i is:

$$R_i^{C1} = \max_{q=0,\ldots,Q_i^{C1}-1} \left(w_{i,q}^{C1} + J_i + C''_i - q \times T_i \right) \quad (6)$$

where C''_i is defined by Equation (5) and T_i is the periodicity of message stream m_i. In this new formulation, erroneous transmissions can occur that imposes longer delays. Then, the overhead of an error recovery mechanism should be accounted for and formulated as:

$$w_{i,q}^{C1} = \left(q + \sum_{j \in hp(i)} \left\lceil \frac{w_{i,q}^{C1} + J_j + Q_{bit}}{T_j} \right\rceil \right) \times P_s + E\left(w_{i,q}^{C1} + C''_i \right). \quad (7)$$

where the function $E(\cdot)$ is as defined by Equation (4).

For the message stream m_i, the number of message requests Q_i available for transmission before the end of level-*i* BP is:

$$Q_i = \left\lfloor \frac{L_i^{C1} + J_i}{T_i} \right\rfloor + 1 \quad (8)$$

where L_i is the length of the longest level-*i* BP, i.e.:

$$L_i^{C1} = \sum_{j \in hp(i) \cup i} \left\lceil \frac{L_i^{C1} + J_j}{T_j} \right\rceil \times P_s + E\left(L_i^{C1} \right) \quad (9)$$

The derivation of the formulation for the remaining cases follows the same rationale described here, as detailed next.

*b) **Case C2**:* There is an instance of the $hp(i)$ message stream generated in the interval $[t_0 - P_s, t_0)$. This case is rather similar to C1 except that we need to consider more interference imposed by the $hp(i)$ message stream. Therefore, Equation (6) and Equation (8) still hold, but $w_{i,q}$ and L_i need to be reformulated as follows:

$$w_{i,q}^{C2} = q \times P_s + \sum_{j \in hp(i)} \left\lceil \frac{w_{i,q}^{C2} + P_s + J_j + Q_{bit}}{T_j} \right\rceil \times P_s + E(w_{i,q}^{C2} + C''_i) \quad (10)$$

$$L_i^{C2} = \sum_{j \in hp(i) \cup i} \left\lceil \frac{L_i^{C2} + P_s + J_j}{T_j} \right\rceil \times P_s + E\left(L_i^{C2} \right) \quad (11)$$

*c) **Case C3**:* In this case neither $hp(i)$ nor $lp(i)$ messages occur during the interval $[t_0 - P_s, t_0)$, but an instance q of the message stream m_i is generated in $[t_0 - P_s, t_0)$, slightly after the synch signal's broadcast. Since in WiDOM only nodes with a non-empty ready queue wait to receive the synch signal [27], the instance q misses to participate in the current tournament phase and should wait for the next superframe. As a consequence, this case is similar to C1, but the duration of

Table I
DIFFERENT CASES OF WCRT COMPUTATION. *lp(i)* / *hp(i)* ARE THE SET OF MESSAGE STREAMS WITH PRIORITY LOWER / HIGHER THAN $\mathbf{m}_i$.

Cases	Message in *lp(i)* arrived in $[t_0 - P_s, t_0)$?	Message m_i arrived in $[t_0 - P_s, t_0)$?	Message in *hp(i)* arrived in $[t_0 - P_s, t_0)$?
C1	✗	✗	✗
C2	✗	✗	✓
C3	✗	✓	✗
C4	✗	✓	✓
C5	✓	✗	✗
C6	✓	✗	✓
C7	✓	✓	✗
C8	✓	✓	✓

an extra superframe must be added into Equation (6), resulting in the following formulation:

$$R_i^{C3} = \max_{q=0,\dots,Q_i^{C3}-1} \left(w_{i,q}^{C3} + J_i + C''_i - q \times T_i \right) + P_s \quad (12)$$

*d) **Case C4**:* There is no $lp(i)$ message generated in the interval $[t_0 - P_s, t_0)$, but both $hp(i)$ and an instance q of the message stream m_i are generated during such interval. With the same reasoning as in C2, Equation (7) and Equation (9) are rewritten as follows:

$$w_{i,q}^{C4} = \left(q + \sum_{j \in hp(i)} \left\lceil \frac{w_{i,q}^{C4} + P_s + J_j + Q_{bit}}{T_j} \right\rceil \right) \times P_s + E\left(w_{i,q}^{C4} + C''_i \right) \quad (13)$$

$$L_i^{C4} = \sum_{j \in hp(i) \cup i} \left\lceil \frac{L_i^{C4} + P_s + J_j}{T_j} \right\rceil \times P_s + E\left(L_i^{C4} \right) \quad (14)$$

Then R_i^{C4} is given by:

$$R_i^{C4} = \max_{q=0,\dots,Q_i^{C4}-1} \left(w_{i,q}^{C4} + J_i + C''_i - q \times T_i \right) + P_s \quad (15)$$

*e) **Case C5**:* Only one instance of $lp(i)$ is generated in the interval $[t_0 - P_s, t_0)$ and there are no other m_i nor $hp(i)$ messages within the same period. As for C2, the new expressions for $w_{i,q}$ and L_i can be formulated as follows:

$$w_{i,q}^{C5} = \left(q + \sum_{j \in hp(i)} \left\lceil \frac{w_{i,q}^{C5} + J_j + Q_{bit}}{T_j} \right\rceil + 1 \right) \times P_s + E\left(w_{i,q}^{C5} + C''_i \right) \quad (16)$$

$$L_i^{C5} = \left(1 + \sum_{j \in hp(i) \cup i} \left\lceil \frac{L_i^{C5} + J_j}{T_j} \right\rceil \right) \times P_s + E\left(L_i^{C5} \right) \quad (17)$$

*f) **Case C6**:* In this case both $hp(i)$ and $lp(i)$ messages are generated in the interval $[t_0 - P_s, t_0)$. Since the low priority message is suppressed by the higher priority one, this case reduces to C2 where there is no $lp(i)$ message instance.

*g) **Case C7**:* There is no $hp(i)$ message in the interval $[t_0 - P_s, t_0)$, but both $lp(i)$ message and an instance q from message stream m_i are generated during this time interval. Clearly, the $lp(i)$ message cannot impose any blocking onto the response time formulation; thus, C7 reduced to C3.

*h) **Case C8**:* An instance of $hp(i)$, $lp(i)$ and the message stream m_i are generated in the interval $[t_0 - P_s, t_0)$. By the same rationale as in the last two cases, $lp(i)$ message is suppressed by higher priority messages and C8 reduces to C4.

To sum up, by looking at the formulation of each case and observing that the response time in C4 is not smaller than those of C1, C2 and C3, the WCRT of a message stream m_i is finally given by:

$$R_i = \max\left(R_i^{C4}, R_i^{C5} \right). \quad (18)$$

V. Experimental Evaluation

In this section, the prior version of Slotted WiDOM protocol is evaluated against the implementation of the novel and more reliable version: the Ack-enabled Slotted WiDOM[1]. We shall use WiDOM interchangeably with Slotted WiDOM, if not stated differently. To the best of our knowledge, this is the first reported implementation of WiDOM (either slotted or unslotted) with an acknowledgment mechanism.

In our experiments, a Wireless Sensor Network (WSN) is composed of 10 MicaZ motes [30] equipped with the WiFLEX add-on board, and generates periodic data traffic (i.e., sensors' measurements) according to the settings shown in Table II. Another MicaZ mote acts as a gateway, which collects data and sends back Ack packet in a single-hop network. An additional MicaZ node is used as interferer, to generate periodic and sporadic bursts of noise.

The periodic interferer generates noise bursts with two distinct inter-arrival times: 70 ms to emulate a Heavy Noisy Channel (HNC) condition, and 200 ms, for a Light Noisy Channel (LNC) condition. The sporadic interferer generates bursts of noise from a two-state Markov model, i.e., "clear channel" and "interference", where the interval of two consecutive bursts of sporadic noise is a random number in the range of [70, 1000] ms. This setting is referred to as Sporadic Noisy Channel (SPNC) condition. The duration of noise bursts in all conditions are equal to one P_s period (due to space limitations more details are provided in [27]). The constraint in Equation (1) implies that the periodicity of the synch signal, P_s, should be larger than 9747 μs [27]. We assume $P_s = 15$ ms, i.e., the master node sends a 300 μs-long synch signal every 15 ms[2].

Our evaluation focuses on five network performance indices, defined as follows:

(*i*) **Packet loss ratio (PLR)**: the ratio of the data (measurements) missed on the gateway over the whole amount of data generated by the sending nodes.

(*ii*) **Deadline miss ratio (DMR)**: the ratio of messages that missed their deadlines over the total number of generated messages. In every scenario, the deadline is implicitly considered to be equal to each message period, which varies from one node to another.

(*iii*) **WCRT miss ratio (WMR)**: the ratio of messages transmitted after their calculated WCRT, over the total number of generated messages.

(*iv*) **Average response time**: the average of all messages' response times received at the gateway.

(*v*) **Energy consumption**: the average energy consumption for each node, expressed as a function of the number of packets exchanged. Experimental trials were performed for both the original and the Ack-enabled versions of Slotted WiDOM, under the different noise conditions. Each run lasted

[1]Due to space constraints, the implementation and setup on the real testbed are not detailed here. Interested readers can refer to [27].

[2]This choice is to allocate time for the gateway to accomplish the data extraction from any received packet and to format the Ack packet.

Table II
SOURCE NODE CONFIGURATION (WITH $P_s = 15\,ms$).

ID (i)	1	2	3	4	5	6	7	8	9	10
$T_i(ms)$	70	180	350	700	1200	1900	3700	5400	5400	5400
$Datarate(bps)$	14629	5689	2926	1463	853	539	277	190	190	190

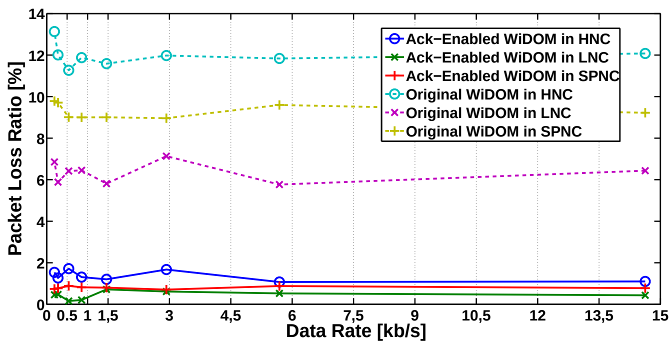


Fig. 5. Packet loss ratio of Ack-enabled WiDOM and original WiDOM.

for 40 minutes, corresponding to 40000+ requests of message transmission.

A. Packet Loss Ratio

Fig. 5 depicts the PLR for both versions of protocols, under different noise conditions. In all scenarios, the Ack-enabled WiDOM largely outperforms the original WiDOM. This is thanks to the Ack packets: they provide a legitimate feedback about the transmission status back to the sender, giving it a chance to retransmit the lost packets (provided that there is enough time left in the superframe for such retransmission [27]). Fig. 5 also confirms that the HNC jeopardizes the network performance more than the other cases. In particular, the PLR under SPNC is lower than that under HNC for both WiDOM versions. Two reasons contribute to this: (i) in heavy noisy channels there is a higher risk of collision for both data and Ack packets; and (ii) a higher collision rate inherently leads to more retransmission requests in the case of Ack-enabled WiDOM, which increases the traffic load.

B. Timing Behavior

Fig. 6 shows the response time achieved by the experimental setup of WiDOM under HNC, LNC and SPNC interference conditions. In particular, three distinct metrics are investigated.

The first metric is the DMR: no deadline miss occurred in all scenarios and for any message stream. This further demonstrates that WiDOM is suited for time-critical applications.

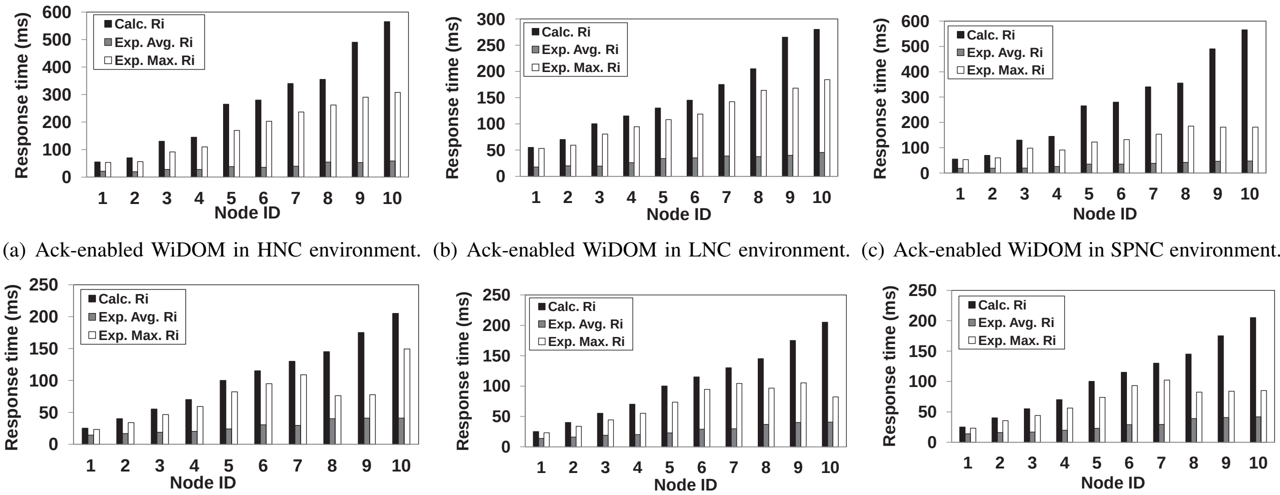
The WMR is the second time metric considered. The analytically computed response time is shown as $Calc.R_i$ in Fig. 6. It has been computed according to the analytical formulation derived in Section IV. The maximum and average values for the response time observed in the tests for each message stream are given by $Exp.Max.R_i$ and $Exp.Avg.R_i$, respectively. The values of $Calc.R_i$ and $Exp.Max.R_i$ confirm that in all scenarios no message experiences a WCRT higher than the estimated one: this substantiates the fact that our model provides a valuable upper bound to the WCRT.

The average value of response time, $Exp.Avg.R_i$, is the third and last metric considered in the timing category. It clearly shows how the Ack-enabled mechanism slightly increases the response time for every message stream. As expected, this is what the system pays to achieve the good results in terms of PLR, as shown in Fig. 5. While the original WiDOM is almost indifferent to the various noise patterns, from Fig. 6 it is clear that heavier noise leads to larger response times for the Ack-enabled Slotted WiDOM. These results are not surprising since, as the noise density increases, there are more packet collisions and consequently a higher number of retransmissions. Accordingly, for the original WiDOM the average response time of message streams does not vary significantly with the channel conditions. The reason is that there is no mechanism devised in the original WiDOM that takes into account the noise of the channel: nodes send their packet according to their arrival schedules, but there is no feedback to let them know about their transmission status.

The last observation suggested us to run another experiment for the original WiDOM in a non-lossy environment, in order to push the system to its limits. Since the original WiDOM does not require extra time for the gateway to process the received data and to issue Ack packets, the synchronization period can be safely decreased from $P_s = 15$ ms to $P_s = 10$ ms. Accordingly, the data rate of the first five message streams can be increased as shown in Table III. Fig. 7 shows the timing behavior in such a non-lossy environment. Once again, no packet loss has been observed, nor deadline miss or WCRT. Indeed, this proves that our framework is able to offer a valid upper bound to the response time of the message streams, under the assumption that the worst channel conditions are known in terms of noise burst duration.

Observing the results given in Fig. 7 it is also evident that a reduction in the synchronization period leads to a slightly better timing behavior, while accommodating nodes with higher data rates. We can conclude that in non-lossy (i.e., interference free) environments or for non-loss sensitive applications (as the case of some less constrained industrial scenarios), the designer can opt for the original Slotted WiDOM approach.

Fig. 8 summarizes the previously reported findings, by showing the average response time as a function of the data rate, under the various interference scenarios. The average response time in the Ack-enabled WiDOM is confirmed to be larger than that for the original WiDOM protocol in all scenarios. In fact a heavier noisy channel deteriorates more the average response time as compared to a lighter one for the Ack-enabled WiDOM, while the average response time



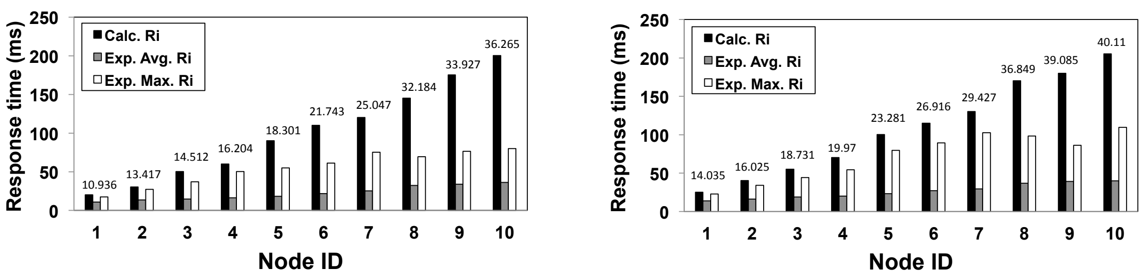
(a) Ack-enabled WiDOM in HNC environment. (b) Ack-enabled WiDOM in LNC environment. (c) Ack-enabled WiDOM in SPNC environment.

(d) Original WiDOM in HNC environment. (e) Original WiDOM in LNC environment. (f) Original WiDOM in SPNC environment.

Fig. 6. Response time comparison of Ack-enabled WiDOM and original WiDOM in different noise environments.

Table III
SOURCE NODE CONFIGURATION (WITH $P_s = 10\,ms$).

ID (i)	1	2	3	4	5	6	7	8	9	10
$T_i(ms)$	30	70	120	300	900	1900	3700	5400	5400	5400
$Datarate(bps)$	34133	14629	8533	3413	1138	539	277	190	190	190



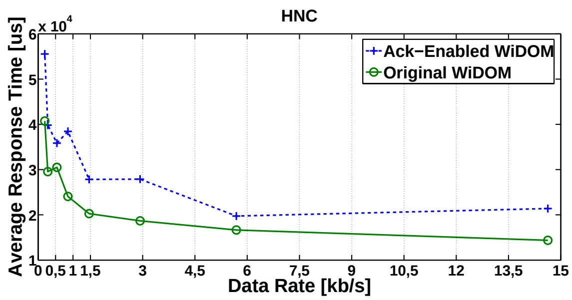
(a) Original WiDOM with $P_s = 10$ ms and setting of Table III. (b) Original WiDOM with $P_s = 15$ ms and setting of Table II.

Fig. 7. Response time comparison of original WiDOM with different P_s. The numbers indicate the value of the average response time.

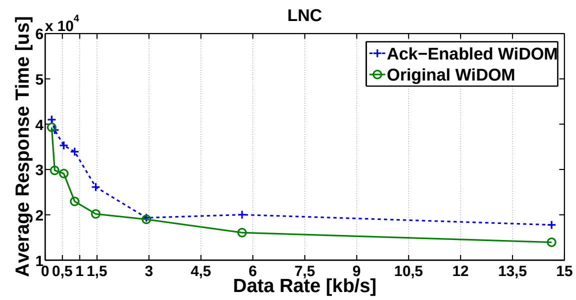
for the original WiDOM does not change significantly under the various noise conditions. Finally, when the environment is non-lossy, smaller average response time values are enabled by the original WiDOM protocol, thanks to the reduction of the synchronization period.

C. Energy Cost

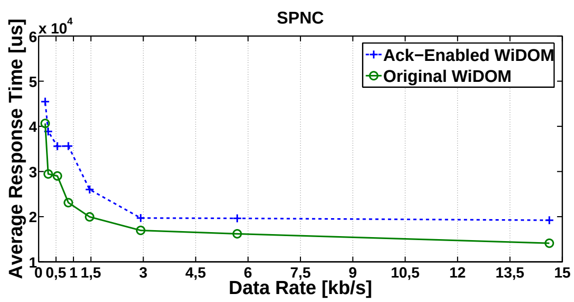
As shown, the Ack mechanism increases the reliability of the WiDOM protocol and its robustness in noisy environments, despite an increase in the response time as well. Nevertheless, for the sake of completeness, the energy cost due to the exchange of extra packets should be accounted for, as a result of (*i*) the amount of energy for receiving the Ack packet and (*ii*) the retransmission, in case of unsuccessful attempt (the energy costs due to the tournament and synchronization phases, which are in charge of the WiFLEX board activity, are common to WiDOM and Ack-enabled WiDOM, thus they have been neglected in the overall computation). Considering the current consumptions of a MicaZ mote [30], i.e., 19.7 mA in RX and 14 mA in TX (for a TX power level of -5 dBm), the energy consumption normalized to the transmission cost for the Ack-enabled WiDOM is estimated as $E[\mathrm{WiDOM^A}] = 3 \times \left[(\mathrm{TX_N} + \mathrm{ReTX_N}) \times 128 + \mathrm{Ack_N} \times 17 \times \frac{19.7}{14}\right]$ where the factor 3 takes into account the voltage supply of the MicaZ node, which is powered by two regular AA batteries, $\mathrm{TX_N}$ is the number of 128 bytes-long transmitted packets, $\mathrm{ReTX_N}$ represents the number of retransmitted packets and $\mathrm{Ack_N}$ is the number of 17 bytes-long Ack packets received by the nodes. Similarly, the normalized energy consumption of the original WiDOM can be estimated as $E[\mathrm{WiDOM^O}] = 3 \times (\mathrm{TX_N} \times 128)$, so, the energy loss ratio, ELR, i.e., the extra energy needed by the Ack-based scheme for the same number of transmission requests, is:



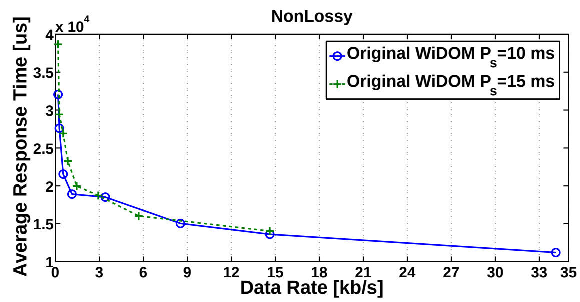
(a) HNC environment.



(b) LNC environment.



(c) SPNC environment.



(d) Original WiDOM in non-lossy environment with different data rates and $P_s = 15$ ms and $P_s = 10$ ms.

Fig. 8. Average response time for Ack-enabled and original WiDOM.

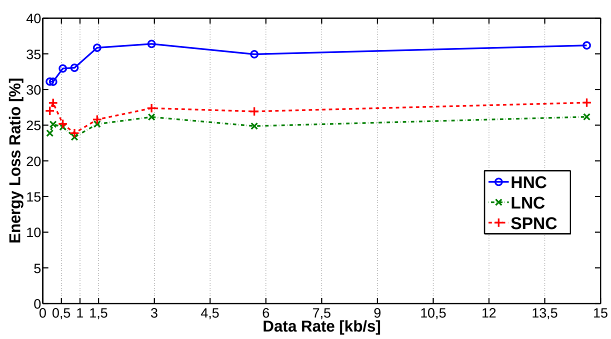


Fig. 9. Energy loss ratio for Ack-enabled WiDOM vs. original WiDOM in HNC, LNC and SPNC environments.

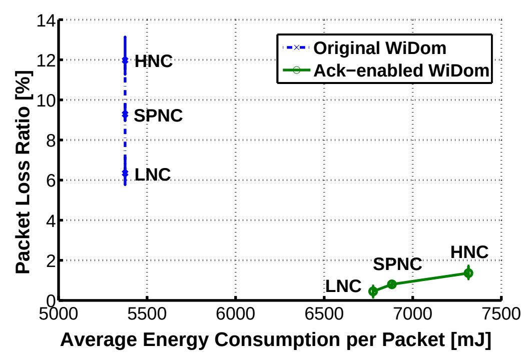


Fig. 10. Packet loss ratio vs. average energy cost per packet for Ack-enabled WiDOM and original WiDOM under HNC, LNC and SPNC environments.

$$\begin{aligned}\mathrm{ELR}[\%] = \quad & 100 \times \frac{E[\mathrm{WiDOM^A}] - E[\mathrm{WiDOM^O}]}{E[\mathrm{WiDOM^O}]} \\ = \quad & 100 \times \frac{\mathrm{ReTX_N} \times 128 + \mathrm{Ack_N} \times 17 \times \frac{19.7}{14}}{128 \times \mathrm{TX_N}}\end{aligned} \tag{19}$$

We do not consider the extra cost paid by the gateway node to receive retransmissions and send back the Ack packets. This makes sense under the common assumption that the gateway is provided with a continuous energy supply.

As it is shown in Fig. 9, the extra energy cost of using the Ack-enabled mechanism compared with the original WiDOM roughly ranges between 25% and 35%, for any data rate. In particular, the ELR shows some fluctuations at lower data rates and tends to stabilize as the data rate increases. The reason is that nodes at lower data rates may have the opportunity to experience disruptive bursts of interference more often than the others, and therefore they might need to retransmit more frequently to successfully deliver their messages. In terms of the interference patterns, it is evident that the higher noise density conditions impose higher energy consumptions due to the higher number of retransmissions. The ELR is about 10% higher for the HNC scenario as compared to the LNC and SPNC cases.

Finally, to further examine the impact of the Ack-based mechanism, the PLR is plotted against the energy consumption per packet in Fig. 10. The energy consumption has been computed using the expressions for $E[\mathrm{WiDOM^A}]$ and $E[\mathrm{WiDOM^O}]$ as above, normalized with respect to the number of transmitted packets, instead of the transmission current. As expected, the energy consumption of WiDOM is not affected by the interference level, while in the case of the

Ack-enabled WiDOM is. This supports our claim that at the cost of roughly 30% more energy, it is possible to achieve a reduction of 90% in the packet loss ratio, i.e., keep it below (2%), regardless of the considered interference.

VI. Concluding Remarks

In this paper, we focused on a prioritized MAC protocol, WiDOM, recently proposed for Wireless (Sensor) Networks and in particular on its recently introduced low-overhead implementation, labeled Slotted WiDOM.

Our primary contribution is the derivation of an accurate worst-case response time analysis for the Slotted WiDOM, where message streams suffer from release jitter and are transmitted in noisy channels. This analysis is non-trivial, due to the slotted nature of the protocol. Moreover, we also proposed an error recovery scheme, which makes Slotted WiDOM more robust and reliable.

To validate our analytical formulation of the worst-case response time, we have conducted a set of experiments under different channel conditions. Besides proving correct that the developed model offers a valid upper-bound to the message streams' response time, experimental results have demonstrated that the Ack-enabled Slotted WiDOM reduces the packet loss rate remarkably, to less than 2%, in all considered noise conditions, and this is an important contribution to foster the use of WiDOM for applications with real-time and quality of service requirements, such as in industrial environments.

As part of our future work, we plan to improve the scalability of the Ack-enabled Slotted WiDOM, by leveraging on the clustering technique to support multi-hop network topologies, where the activities of star-based clusters can be scheduled according to the approach proposed in [31], [32]. We also aim to extend the experimental evaluation in a large scale industrial plant to obtain the worst case response time of message streams under even more realistic noise patterns.

Acknowledgments

This work was partially supported by National Funds through FCT/MEC (Portuguese Foundation for Science and Technology) and when applicable, co-financed by ERDF (European Regional Development Fund) under the PT2020 Partnership, within project UID/CEC/04234/2013 (CISTER Research Centre); also by FCT/MEC and ERDF through COMPETE (Operational Programme 'Thematic Factors of Competitiveness'), within project(s) FCOMP-01-0124-FEDER-028990 (PATTERN), by FCT/MEC and the EU ARTEMIS JU within project ARTEMIS/0001/2013 - JU grant nr. 621429 (EMC2).

References

[1] L. Sha and S. Sathaye, "A systematic approach to designing distributed real-time systems," *IEEE Comp. J.*, vol. 26, no. 9, pp. 68 –78, 1993.
[2] R. Obenza, "Rate monotonic analysis for real-time systems," *IEEE Comp. J.*, vol. 26, no. 3, pp. 73 –74, 1993.
[3] R. Bosch, "CAN specification v2.0," *Rober Bousch GmbH, Postfach*, vol. 300240, 1991.
[4] B. Andersson and E. Tovar, "Static-priority scheduling of sporadic messages on a wireless channel," in *IEEE Conf. on Principles of Distr. Sys.*, pp. 322–333, 2005.
[5] N. Pereira et al., "WiDom: A dominance protocol for wireless medium access," *IEEE TII*, vol. 3, no. 2, pp. 120 –130, 2007.
[6] B. Andersson et al., "A scalable and efficient approach for obtaining measurements in CAN-based control systems," *IEEE TII*, vol. 4, no. 2, pp. 80–91, 2008.
[7] N. Pereira et al., "Efficient aggregate computations in large-scale dense WSN," in *IEEE RTAS Conf.*, pp. 317 –326, April 2009.
[8] C. A. Boano et al., "Jamlab: Augmenting sensornet testbeds with realistic and controlled interference generation," in *ACM/IEEE IPSN Conf.*, pp. 175–186, 2011.
[9] D. Maxim and L. Cucu-Grosjean, "Response time analysis for fixed-priority tasks with multiple probabilistic parameters," in *IEEE RTSS*. pp. 224–235, 2013.
[10] J. Song et al., "WirelessHART: Applying wireless technology in real-time industrial process control," in *IEEE RTAS Conf.*, pp. 377–386, 2008.
[11] H. Trsek et al., "A flexible approach for real-time wireless communications in adaptable industrial automation systems," in *IEEE ETFA Conf.*, pp. 1–4, 2011.
[12] W. Shen et al., "PriorityMAC: A priority-enhanced MAC protocol for critical traffic in industrial wireless sensor and actuator networks," *IEEE TII*, vol. 10, no. 1, pp. 824–835, 2014.
[13] S. X. Ding et al., "An integrated design framework of fault-tolerant wireless networked control systems for industrial automatic control applications," *IEEE TII*, vol. 9, no. 1, pp. 462–471, 2013.
[14] P. Gaj et al., "Computer communication within industrial distributed environments: a survey," *IEEE TII*, vol. 9, no. 1, pp. 182–189, 2013.
[15] I. Aad and C. Castelluccia, "Differentiation mechanisms for IEEE 802.11," in *INFOCOM IEEE Conf.*, vol. 1, pp. 209–218, 2001.
[16] J. L. Sobrinho and A. S. Krishnakumar, "Real-time traffic over the IEEE 802.11 medium access control layer," *Bell Labs Tech. J.*, vol. 1, no. 2, pp. 172–187, 1996.
[17] B. P. Crow et al., "IEEE 802.11 wireless local area networks," *IEEE Comm. Mag.*, vol. 35, no. 9, pp. 116–126, 1997.
[18] M. Barry et al., "Distributed control algorithms for service differentiation in wireless packet networks," in *INFOCOM IEEE Conf.*, vol. 1, pp. 582–590, 2001.
[19] D. Jiunn and R.-S. Chang, "A priority scheme for IEEE 802.11 DCF access method," *IEICE Trans. on Comm.*, vol. 82, no. 1, pp. 96–102, 1999.
[20] "Industrial communication networks–wireless communication network and communication profiles–WirelessHART," *IEC Standard, 62591*, 2010.
[21] J. Sobrinho and A. S. Krishnakumar, "Quality-of-service in ad hoc carrier sense multiple access wireless networks," *IEEE JSAC*, vol. 17, no. 8, pp. 1353–1368, 1999.
[22] "IEEE standard 802.15.4, part. 15.4: Wireless medium access control (MAC) and physical layer (PHY) specification for low-rate wireless personal area networks (LR-WPANs)," *IEEE Standard*, 2006.
[23] "IEEE standard for local and metropolitan area networks–part. 15.4: Low-rate wireless personal area networks (LR-WPANs)–amendment 1: MAC sublayer," *IEEE Standard 802.15.4e*, 2012.
[24] A. Koubâa et al., "i-GAME: an implicit GTS allocation mechanism in IEEE 802.15.4 for time-sensitive wireless sensor networks," in *ERTS Conf.*, pp. 192–202, 2006.
[25] "Wireless systems for industrial automation: Process control and related applications," *ISA100.11a Standard*, 2009.
[26] "Industrial communication networks–fieldbus specifications–WIA-PA communication network and communication profile," *IEC Standard*, 2011.
[27] "Implementation details of Ack-enabled Slotted WiDOM," 2014. [Online]. Available: https://goo.gl/Q8fr17
[28] A. K. Mok and S. Ward, "Distributed broadcast channel access," *Comp. Net.*, vol. 3, pp. 327–335, 1979.
[29] R. Davis et al., "Controller area network (CAN) schedulability analysis: Refuted, revisited and revised," *Real-Time Sys.*, vol. 35, pp. 239–272, 2007.
[30] Crossbow Inc, *Datasheet: MICAz*, San Jose, USA, 2004.
[31] S. Tennina et al., *IEEE 802.15.4 and ZigBee as enabling technologies for low-power wireless systems with quality-of-service constraints*. Springer Science & Business Media, 2013.
[32] A. Koubâa et al., "TDBS: a time division beacon scheduling mechanism for ZigBee cluster-tree wireless sensor networks," *Real-Time Sys.*, vol. 40, no. 3, pp. 321–354, 2008.